



Ciclo de Vida Propuesto del Proyecto

Proyecto: ByteBuss

Equipo de Desarrollo: Los pozoles

Caro

Raúl

Zacek

Edgar

Ciclo de Vida Propuesto del Proyecto

Introducción

El **ciclo de vida del proyecto ByteBuss** se diseñó bajo un **enfoque ágil**, adoptando el marco de trabajo **Scrum**, ideal para proyectos complejos que requieren **flexibilidad, colaboración y entregas incrementales de valor**.

Este enfoque permite **dividir el desarrollo en sprints cortos**, validar avances con el cliente de forma continua y **adaptarse a cambios** de forma rápida y controlada.

Modelo de Desarrollo Adoptado: SCRUM

Scrum es un marco **iterativo e incremental** que divide el proyecto en **sprints** (ciclos cortos de trabajo de 1-2 semanas).

Principios clave:

- Transparencia y visibilidad del progreso.
- Entregas frecuentes e incrementales.
- Validación temprana con el cliente.
- Colaboración y retroalimentación constante.
- Mejora continua.

Fases del Ciclo de Vida del Proyecto

◆ Inicio del Proyecto

- Definir **alcance y objetivos estratégicos**.
- Identificar **stakeholders** y sus expectativas.
- Elaborar el **Product Backlog inicial**.
- Asignar **roles y responsabilidades**.
- Selección de **arquitectura (MVC)** y herramientas principales.

Resultado: Visión compartida, objetivos claros y backlog inicial priorizado.

◆ Planificación de Sprints

- Dividir el trabajo en **sprints** de 1-2 semanas.
- Priorizar historias de usuario con el cliente.
- Estimar esfuerzo y definir criterios de aceptación.
- Elaborar el **Sprint Backlog**.

Resultado: Plan claro y alcanzable por sprint.

◆ Desarrollo Iterativo e Incremental

- Codificación de funcionalidades.
- Diseño UI/UX responsivo y amigable.
- Implementación de APIs RESTful.
- Integración con bases de datos y pagos NFC.
- Control de versiones en GitHub (Git Flow).

Resultado: Incremento funcional entregable y probado.

◆ Pruebas y Control de Calidad

- Pruebas unitarias y de integración automatizadas.
- Validación de criterios de aceptación.
- Gestión de incidencias en Jira/Trello.
- Uso de GitHub Actions para CI/CD.

Resultado: Incremento estable y listo para producción.

◆ Revisión y Validación con el Cliente

- **Sprint Review:** presentación de avances.
- Revisión de funcionalidades entregadas.
- Recepción de retroalimentación.

- Ajuste del Product Backlog.

Resultado: Validación parcial con el cliente y ajuste continuo.

◆ **Despliegue y Entrega**

- Preparar ambientes de prueba y producción.
- Contenedorización con Docker, balanceo con Nginx.
- Despliegue en la nube (AWS/Azure).
- Documentación técnica y manuales de usuario.
- Capacitación a operadores y usuarios finales.

Resultado: Sistema funcional y listo para uso real.

◆ **Mantenimiento y Soporte**

- Resolución de bugs y fallos reportados.
- Monitoreo (logs, métricas, dashboards).
- Actualizaciones y mejoras continuas.
- Soporte Help Desk para usuarios finales.

Resultado: Operación estable, soporte continuo y confianza del cliente.

Cronograma Tentativo

Sprint	Duración	Objetivos Principales
Sprint 1	Semanas 1-2	Alcance, planificación, arquitectura, repositorio base.
Sprint 2	Semanas 3-4	Backend base, modelo de datos, conexión a base de datos.
Sprint 3	Semanas 5-6	Frontend inicial, integración parcial, guía de estilos.

Sprint 4	Semanas 7-8	Funcionalidades clave, integración completa, pruebas unitarias.
Sprint 5	Semanas 9-10	QA completo, corrección de errores, documentación final.

Ventajas del Ciclo de Vida Propuesto

- Entregas **incrementales y continuas** de valor.
- **Adaptación rápida** a cambios en requisitos.
- Validación **temprana y constante** con el cliente.
- **Transparencia y trazabilidad** del avance.
- Reducción de **riesgos y costos** de retrabajo.
- Fomento de **colaboración** y mejora continua.